Modeling complex particles phase space with GAN for Monte Carlo SPECT simulations: a proof of concept

David Sarrut, A. Etxebeste, Nils Krah, Jean Michel Létang

Modeling complex particles phase space with GAN for Monte Carlo SPECT simulations: a proof of concept

D. Sarrut[1], A. Etxebeste[1], N. Krah[1,2], JM. Létang[1]

[1] Université de Lyon, CREATIS; CNRS UMR5220; Inserm U1044; INSA-Lyon; Université Lyon 1; Centre Léon Bérard, France.

[2] University of Lyon, Université Claude Bernard Lyon 1, CNRS/IN2P3, IP2I Lyon, F-69622, Villeurbanne, France

E-mail: david.sarrut@creatis.insa-lyon.fr

Abstract.

A method is proposed to model by a Generative Adversarial Network the distribution of particles exiting a patient during Monte Carlo simulation of emission tomography imaging devices. The resulting compact neural network is then able to generate particles exiting the patient, going towards the detectors, avoiding costly particle tracking within the patient. As a proof of concept, the method is evaluated for SPECT imaging and combined with another neural network modeling the detector response function (ARF-nn). A complete rotating SPECT acquisition can be simulated with reduced computation time compared to conventional Monte Carlo simulation. It also allows the user to perform simulations with several imaging systems or parameters, which is useful for imaging system design.

1. Introduction

Monte Carlo simulations in medical physics are widely used in the design and development of imaging systems such as positron emission tomography (PET) or single photon emission computed tomography (SPECT), to monitor nuclear decay, fragmentation in the patient body or for range verification in particle therapy. For example, many works on emerging instrumentation for SPECT imaging systems [1, 2, 3] require extensive and realistic Monte Carlo simulations to investigate and optimize the detection modules and novel geometrical configurations such as multi-head detectors. In abstract terms, such simulations create a mapping from a given source distribution inside the patient to a signal captured by the imaging device outside of the patient by transporting particles one-by-one through the objects present in the simulation. Because some of these objects do typically not overlap, it is possible to decompose the entire simulation into intermediate steps. For example, in the Monte Carlo simulation of a SPECT imaging system, a first step transports particles through the patient anatomy described, e.g., by a CT image and a second step transports those particles exiting the patient to and through the detector system. During the first step, photons emitted from an activity distribution of a given radionuclide are tracked in the inhomogeneous medium, potentially undergoing Compton scattering, until they are absorbed or exit the medium. The second step involves the simulation of the photon interactions within the detection head, through the collimator and the scintillator.

Decomposing a simulation is useful to avoid redundancy in certain applications. For example, in a given SPECT scanner, the imaging device is always identical and only the patient anatomy in the first step changes. In this case, the explicit transport of particles across the imaging device can be replaced by a collimator–detector (angular) response function (ARF) that combines the accumulated effects of all interactions in the imaging head. The ARF may be approximated by an effective numerical model, provides variance reduction and accelerates the simulation [4, 5, 6]. On the other hand, e.g., when studying different imaging system designs, only the second simulation step needs to be repeated while the first step, i.e. the transport across a given patient, remains

unchanged. This requires a way to store or model the result of the first simulation step and this paper proposes a method to achieve that.

More specifically, we focus on the emission and transport of gammas in the patient (described by a CT image). The phase space parameters (position, momentum, and energy) of all particles exiting the patient provide sufficient information to serve as a source description for a subsequent simulation or as input to ARF. The phase space dataset can in principle be stored in a file and reused repeatedly later. This is e.g. a commonly used method for the simulation of Linac treatment heads where particles are transported from the electron beam hitting the tungsten target to the different head elements to finally be registered in a virtual plane at the exit of the head [7]. A disadvantage is that those files are generally large (several GB) and can be cumbersome to process, use and exchange, which is particularly relevant when simulating a complete SPECT acquisition with potentially billions of particles to be transported. Several works, such as in [8, 9, 10], provided methods to model accelerator phase space distributions analytically, but they have never been investigated for SPECT simulations.

In this work, we propose and explore the use of a generative model to describe the phase space distribution of particles exiting the patient volume in SPECT Monte Carlo simulation. Specifically, we rely on the concept of Generative Adversarial Networks (GAN) which have the potential to model multidimensional probability distributions [11]. One component of the GAN, i.e. a neural network called generator G, serves as a compact and fast source of particles for the Monte Carlo simulation. Previous work has shown that the phase space of particles exiting a Linac head can be modeled with a GAN trained through analog Monte Carlo simulation [12]. The phase space distributions of particles in that work were overall relatively smooth. Non-smooth features in the distributions (e.g. photo-peaks) could not be correctly modeled by the GAN.

In this paper, we propose to explore and extend the concept to a more complex phase space: the phase space of particles exiting a patient in a 3D SPECT acquisition. The goal is to develop a GAN which is able to model the distribution of the exiting gammas so that they can be generated without the need to track them (again) within the patient. Moreover, we show that it is possible to combine this GAN with another neural network that models the detector response, as proposed in [6].

2. Material and methods

Following the approach described in [12], the proposed method is split into 3 main steps: 1) generate the training dataset via Monte Carlo simulation, 2) train the GAN and 3) use the generator of the GAN as a source. In the following, a second neural network (ARF-nn), is used to model the imaging detector response and the two neural networks are combined. ARF-nn stands for neural network-based Angular Response Function, proposed in [6], which models the detector response. The general principle of the proposed concept is illustrated figure 1.

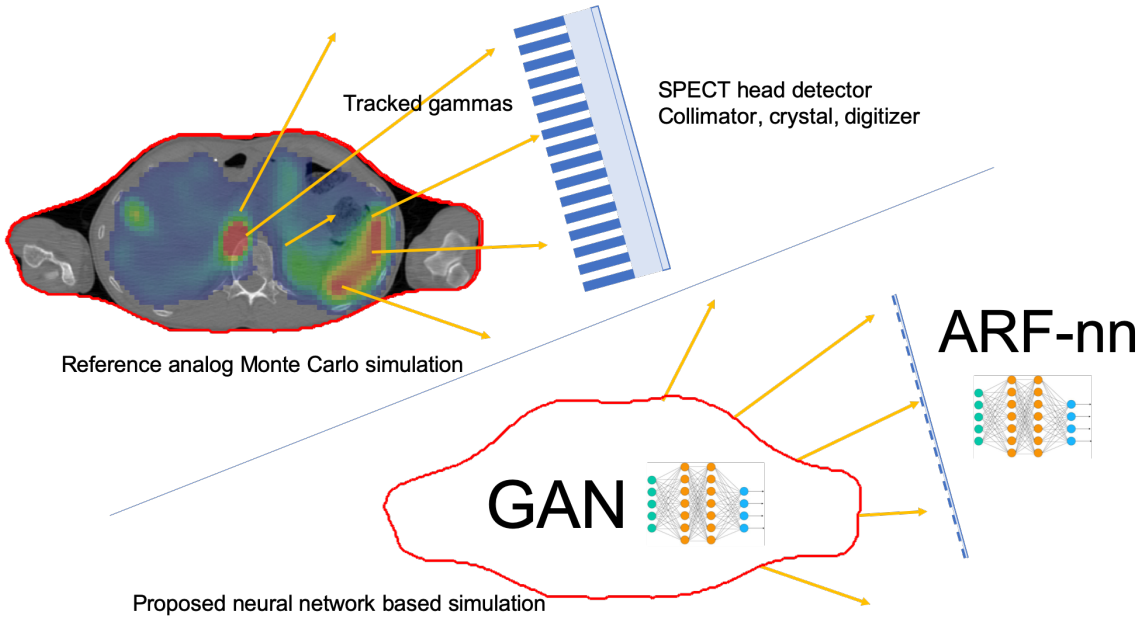

Figure 1. Principles of the combined method. Top row depicts the reference Monte Carlo simulation of a SPECT acquisition, including anatomical image (a Computed Tomography or CT here), voxelized source activity, gammas tracking and SPECT head complete description (collimator, crystal, electronics). The bottom row illustrates the method combining two neural networks: gammas are generated by a GAN, tracked in straight lines to a detector plane, serving as input for the ARF-nn to create the projection multi-channel image.

2.1. Training dataset from Monte Carlo simulation

We considered the simulation of a complete SPECT acquisition. It consisted of a 3D CT image, a 3D activity source and a single SPECT head rotating around the patient. The 3D CT image was described as a matrix of voxels associated with material density and composition following the stoichiometric calibration method [13, 14]. The activity source can be any 3D image where voxels are associated with a known activity in MBq. Without loss of generality, only ^{177}Lu was considered but any other radionuclide may be used. We selected ^{177}Lu because it is currently used for several radionuclide therapy treatments, notably combined with somatostatine analogues or PSMA (neuroendocrine tumors, prostatic adenocarcinoma), and SPECT images are used to monitor the patient dose distribution thorough the treatment. Gammas were emitted isotropically from randomly sampled positions in each voxel following the emission energy spectrum of the radionuclide. The half-life is 159.53 hours. In addition to electrons (max 497 keV, abundance rate of 78.6%), each decay emits around 17.2% of gammas of which 10.3% of 208 keV and 6.2% of 113 keV. The activity injected into the patient is assumed to be 7.4 GBq (typical clinical injection). We assumed the SPECT image to be acquired 24 h after injection leading to 7.07 GBq due to the exponential decay. We considered that only half of this quantity stays in the patient part visible from the camera head due to

the physiological washout (as can be observed in our clinical practice). Without loss of generality we consider that one SPECT angular projection lasts 15 seconds and that a complete acquisition rotation contains 60 projections every 6 degrees. This leads to approximately 5.3×10^{10} decays or 9.1×10^{9} emitted gammas for one single projection neglecting the decay during acquisition, and about 40×10^{6} detected counts. Since the ARF-nn method was used, we only need about 4.6×10^{8} emitted photons to simulate an image with variance equivalent to a real clinical acquisition.

In order to generate a phase space containing the gammas exiting the patient skin, the first step was to define the surface to which phase space information about the gammas refers. The following aspects are to be considered. (1) gammas are tracked in the voxelized CT image until they leave the boundaries of the CT image. (2) In most of clinical devices, SPECT heads rotate around the patient and may move as close as possible to the patient skin. (3) With parallel collimator, the count rate does not change significantly when the patient-to-collimator distance decreases, because it is compensated for by the increased solid angle, but the spatial resolution improves [15]. Hence, the exiting gammas should be stored as close to the patient skin as possible in order to accommodate all possible collimator positions. To this end, an extension of the phase space scorer of Gate (`GatePhaseSpaceActor`) was developed to use a binary mask image as additional parameter (see section 3 for more information about Gate). This mask image was created from the anatomical image (CT here) and is used to store gammas in the phase space as soon as they reach the air volume surrounding the patient's body, thus exiting the patient skin. The mask image (1 inside the patient, 0 outside), is build by extracting the patient contour from the CT thanks to an automated algorithm [16] based on morphological operations. Moreover, it is important to ensure that gammas exiting the skin will not re-enter the patient, as this can be the case for example in the empty space between the thorax side and the arms, when arms are not above the head. To avoid those situations, a large morphological closing operator (60 mm radius) was applied to remove all those types of voids and create a quasi-convex surface, as illustrated in figure 2. Note that the use of this mask does not modify the computation time of the simulation.

It could also be interesting to only store in the phase space the gammas having energy that have a chance to be detected by the SPECT head and low/high energy thresholds can be provided. In that case, the ratio of omitted versus stored gammas should be taken into account in order to correctly scale the simulation. According to the tolerance defined by the user, more restrictive thresholds could be used to further reduce the number of gamma that will be generated.

As a summary for this first step, a Monte Carlo simulation is performed to track emitted gammas through the patient, storing in a phase space file all exiting particles from a surface covering the patient skin. Stored particle information are: energy, 3D position and 3D direction cosines (3D normalized vector of the photon momentum), so seven dimensions. This phase space constitutes the training dataset that will be the input of the GAN (next section). Note that only gammas are considered here, but if

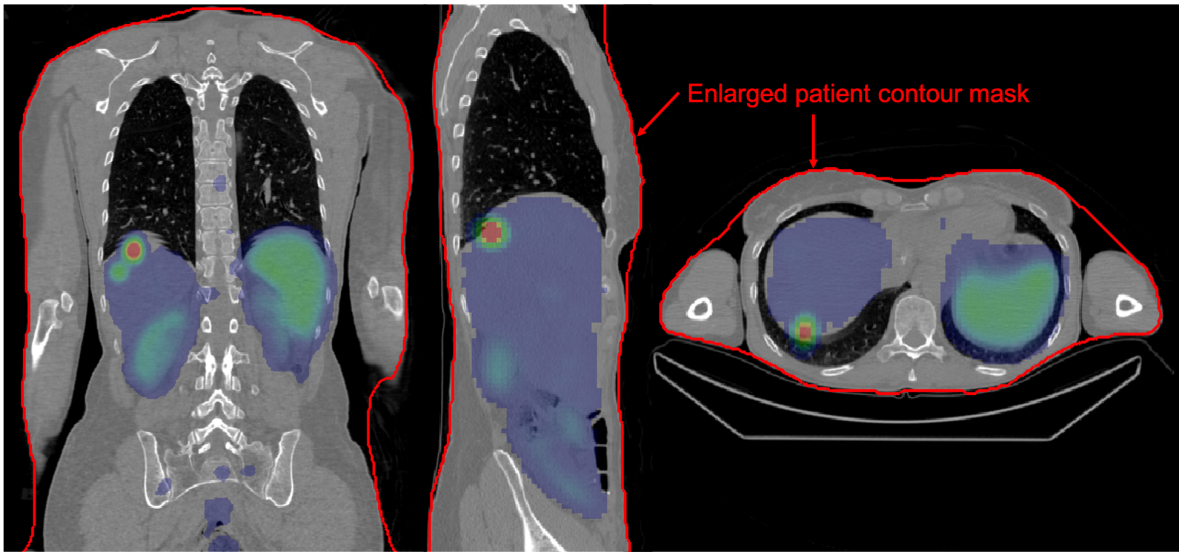

Figure 2. Example of a patient CT slices (coronal, sagittal and axial) used in the simulation overlaid with an activity source obtained from a SPECT image. The patient contour binary mask, enlarged by the closing operator, is shown with red contour. During the simulation, gammas are stored when they reach the outside of the mask (red contour). Note that one can observe artifacts in the CT due to patient breathing motion.

needed, additional types of exiting particles (electron, positron) may also be accounted for.

2.2. Training the GAN

Taking as input the previously described dataset, the goal of training the GAN is to build a generative neural network G able to generate particles following the distribution of the gammas in the training dataset. GAN optimization alternates the interdependent training of two neural networks, the generator G and the critic (or discriminator) D [11]. The proposed GAN architecture model was the following: the Wasserstein GAN loss function, equation 1, proposed in [17] was used together with gradient penalty (GP) [18].

$$\text{WGAN Loss} = \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{x}})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x})] + \lambda \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}[\text{GP}] \tag{1}$$

In the equation, $\mathbb{P}_r$ refers to the (*real*) data distribution and $\mathbb{P}_g$ to the model (*generated*) distribution defined by $\tilde{\mathbf{x}} = G(\mathbf{z})$, with the noise $\mathbf{z}$ following a uniform or a normal distribution, following the notations of [18]. Wasserstein loss is an alternative to previously proposed Kullback-Leibler and Jensen-Shannon divergences to quantify the distance between the data distribution. It is based on the Earth-Mover distance and Optimal Transport theory. It evaluates the cost of the cheapest transport plan between the multidimensional distributions and was shown to provide better stability compared to original GAN. It requires that D be 1-Lipschitz (the norm of its gradients is at most 1 everywhere). Instead of the clipping strategy that was initially recommended

in [17] and used in [12], several gradient penalty strategies have been reported since. Gulrajani et al. [18] proposed to penalize when the gradient differs from 1. The penalty is computed by sampling uniformly along straight lines pairs of samples from the training dataset and from the generator. A hyperparameter λ is used to control the strength of this penalty. Petzka et al. [19] and Thanh-Tung et al. [20] proposed alternative penalties: Square Hinge (or Lipschitz penalty in the article) and 0-GP (GP stands for Gradient Penalty), using maximum and zero-centered penalty instead of the distance to 1. Recently, Jolicoeur-Martineau et al. [21] proposed a unified way to look at those penalties, considering the types of norm used to penalize the gradient (L2 or L^∞) and the loss types (Least Square or Hinge). Table 1 summarizes the considered gradient penalties.

Grad. Pen.	Least Square	Hinge
L1	$(\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_1 - 1)^2$	$\max\{0, (\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_1 - 1)\}$
L2	$(\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_2 - 1)^2$ [18]	$\max\{0, (\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_2 - 1)\}$
L^∞	$(\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_\infty - 1)^2$	$\max\{0, (\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_\infty - 1)\}$
Square Hinge	$(\max\{0, (\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_2 - 1)\})^2$ [19]	
0-GP	$(\|\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|\|_2)^2$ [20]	

Table 1. Gradient penalties according to [21, 18, 20, 19]. In the equations, $\hat{\mathbf{x}} = \alpha\mathbf{x} + (1-\alpha)\mathbf{y}$, with $\mathbf{x}$ sampled from $\mathbb{P}_r$ the *real* probability distribution of the gammas from the training dataset, and $\mathbf{y}$ is sampled from $\mathbb{P}_g$ the *generated* gamma distribution. $\alpha \sim \mathcal{U}(0,1)$ is sampled from the unit hyperball (following notation of [18]).

GAN training stabilization and convergence are still intensively being studied, both theoretically and experimentally. It is still not clear what kind of penalty is better. The RMSProp method [22] was used for the GAN optimization. Learning rates for the G and D networks were fixed experimentally to 10^{-4} and 2×10^{-5}, respectively. The architecture of G and D networks was a fully connected neural network with 4 hidden layers, 700 neurons in each and the activation function was Rectified Linear Unit (ReLu). The total number of weights of both networks was almost 2 million. The number of dimension z of the generator was set to 9. Stochastic batches of 10^4 gammas were used at each iteration. The critic D was updated twice per generator update. The total number of epochs was set to 10^5.

2.3. Combining GAN and ARF-nn

Once the GAN is trained, the generator G can be used in the simulation of a complete SPECT acquisition. For those simulations, the initial CT and the activity source were removed and replaced by G as a source of gammas exiting patient skin and

moving towards the detectors. The SPECT heads consist of two detectors composed of collimator, crystal and a complete digitization chain [23]. The heads were rotated around the patient and acquired incoming gammas for typically 15 seconds, in order to create projection images with one channel per energy window. The neural network-based Angular Response Function (ARF-nn) method was used [6] to model the SPECT head. With this approach, the detector is replaced by a plane and the response function takes as input the energy and direction of the incoming gammas and provides the probability of counts in all energy windows. The response function is a neural network that was trained with an analog Monte Carlo simulation of the full description of the detector. Hence, the two neural networks, G and ARF-nn, were used successively: the gammas generated by G were transported in a straight lines to the ARF plane where they were used as input to the ARF-nn. This approach is only valid if there are no additional objects between the patient contour and the detector. In this paper, the reference simulation was also performed with ARF-nn in order to only evaluate the impact of using GAN.

3. Experiments

Simulations were performed with Gate version 9.0 [24], using Geant4 version 10.6 [25] and PyTorch framework version 10.1 [26] with CUDA GPU acceleration. All experiments used the Geant4 physics list "standard electromagnetic option 4". Production cuts were set to 0.1 mm. Computations were performed on an Intel Xeon CPU E5-2640 v4 @ 2.40 GHz with NVIDIA Titan Xp (GP102-450-A1, 12 GB memory) and on the Jean Zay CNRS computing center (IDRIS, GENCI, Orsay, France).

Throughout this work, the considered SPECT system was the imaging head of the GE Discovery 670 with NaI(Tl) crystal. The real camera is composed of two heads but only one head was considered here. The collimator used for ^{177}Lu was the medium energy general purpose (MEGP) parallel-hole one. The collimator hole diameters were 3 mm with a septal thickness of 1.05 mm and the crystal thickness was 9.525 mm (3/8 inch). The effect of the digitizer chain was modeled by applying a spatial Gaussian blurring of 3.97 mm [27] and an energy resolution of 10 % at 171 keV. The head was replaced by ARF-nn trained to model the detection response as described in [6]. Gantry rotation was performed with constant 30 cm distance between the rotation center and the detector.

3.1. Experiment1: spherical sources

The first experiment was conducted with an artificial source of activity composed of a hot background area and 4 spheres of 40, 30 and 20 mm radius with an activity concentration 20 times and 50 times higher than the background, for a total of 3.5 GBq of ^{177}Lu, as shown in figure 3. The sources were positioned in the thorax region in the CT image of a patient to obtain various attenuation conditions (one source is

in the lung parenchyma, the others in soft tissues). 10^9 primary decays following the energy spectrum of ^{177}Lu were simulated (208.4 keV at 10.4% and 112.9 keV at 6.2% for the two main photopeaks), resulting in about 1.72×10^8 emitted gammas. 60 reference SPECT projections obtained over 360° were generated with 10^9 decays (1.4×10^8 primary particles) per rotation angle. The projections were reconstructed with the method described in [28] and implemented in the RTK toolkit [29], using the OSEM algorithm with quadratic penalization [30], 10 iterations and 15 subsets, 5 mm voxel size matrix. The number of iterations/subsets were chosen empirically. Scatter correction was taken into account through the Double Energy Window method [31]. Attenuation correction (AC) and point spread function (PSF) correction were performed during the iterative process using the method described in [32]. Moreover, the reference simulation was performed 30 times in order to estimate the mean and standard deviation of all pixels of the generated projection image.

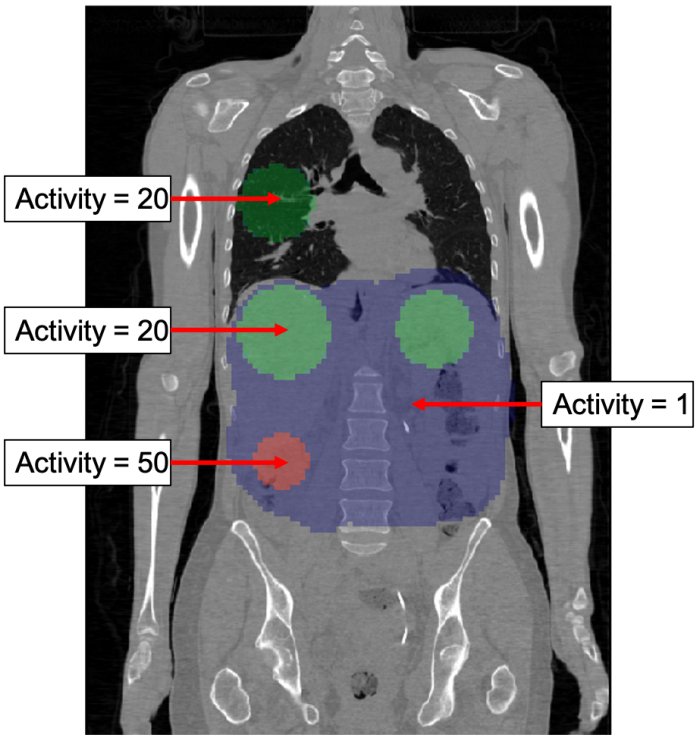

Figure 3. Activity sources (four spheres and background) in a non homogeneous CT image for the Experiment1. The activity concentrations in the spheres are 20 and 50 times higher than the activity in the background.

A simulation with the same source of activity, with only 10^8 primary decays and without the SPECT device was performed to create the training phase space dataset containing the gammas exiting the patient. It was performed with the phase space scorer extension that records every gamma which traverses the patient contour as described previously. Two phase spaces were computed with different random seeds to be used as independent training and validation datasets. The GAN was trained with the first

dataset using the parameters detailed in the previous section. In this experiment the gradient penalty Square Hinge (from [19], see table 1) has been used with $\lambda = 10$. Once the generator was trained, the SPECT projections were generated by the combined method previously described and compared to the reference projections. Six energy windows were used (the two photopeaks and associated adjacent 8% scattering windows, see table 2). Projection images were generated for all six windows. Reference (ARF-nn only) and GAN+ARF-nn were compared based on marginal distribution histograms of the gammas exiting the patient and profiles in the projection images. Projections obtained from the GAN method were also reconstructed with the same algorithm and parameters than the reference.

Energy windows	low	high
Scatter1	96 keV	104 keV
Peak1 113 keV	104.52 keV	121.48 keV
Scatter2	122.88 keV	133.12 keV
Scatter3	176.64 keV	191.36 keV
Peak2 208 keV	192.4 keV	223.6 keV
Scatter4	224.64 keV	243.36 keV

Table 2. Energy windows used during simulations.

3.2. Experiment2: realistic patient activity

The second experiment was performed with a realistic activity source obtained from a SPECT image reconstruction scaled such that the whole source contains 3.5 GBq. Like for the previous experiment, 60 projections over 360° were generated and reconstructed (same parameters). Similarly to Experiment1, a phase space was generated from the activity source with 10^9 primary particles and a GAN was trained from the dataset. Several gradient penalties were compared. Once trained, projections were generated with the proposed method and compared to the reference projections using the Hellinger distance which takes into account the mean and the variance of the detected count values. This distance is computed for all pixels considering that the detected counts follow a Poisson distribution (the mean is equal to the variance). Let c_r be the pixel values in the reference image and c_g in the GAN generated image, the Hellinger distance is computed by equation 2 for all pairs of pixels and averaged for five different angles (every 72 degrees).

$$Hd(c_r, c_g) = 1 - \exp -\frac{1}{2}(\sqrt{c_r} - \sqrt{c_g})^2 \tag{2}$$

4. Results

The phase space files generated for Experiment1 and Experiment2 contained each approximately 1.5×10^8 stored gammas (4 GB of disk space). Approximately 15% of the

emitted gammas are not stored in the phase space (due to attenuation) and about 20% of the stored gammas correspond to scattered gammas which exit the phantom/patient with a reduced energy (different from the two photopeaks). Of course, those values vary according to the activity distribution, the anatomical medium and the radionuclide type. The figure 2 illustrates the initial source of activity for Experiment2.

Figure 4 depicts the values of the GAN loss during the training process, both with the training and the validation datasets. Figure 5 depicts the marginal distributions (150 bins) for the seven dimensions (energy, positions, directions) computed from 10^4 samples obtained from the reference phase space and generated by the generator G of the GAN. The reference phase space shows a step-like structure (especially Z) due to the discrete nature of the mask image where the gammas are recorded. The GAN tends to smooth these steps.

Figures 6 and 7 illustrate projection obtained from the reference and from the GAN method, for different angles (0°, 72°, 222°), for the 208 keV peak energy window and 96-104 keV scatter window. The red lines indicate the location of the profiles in the figure 8. Since the reference simulation was performed 30 times, the mean was used as reference and the standard deviation was used to depict the error bands. The six energy windows (scatters and peaks) of ^{177}Lu are compared. Note that the Y-scale of number of counts is different in each subplot, as it is much larger for the peaks energy windows than for the scatter windows. The number of detected counts in a projection was in the order of 0.3×10^6.

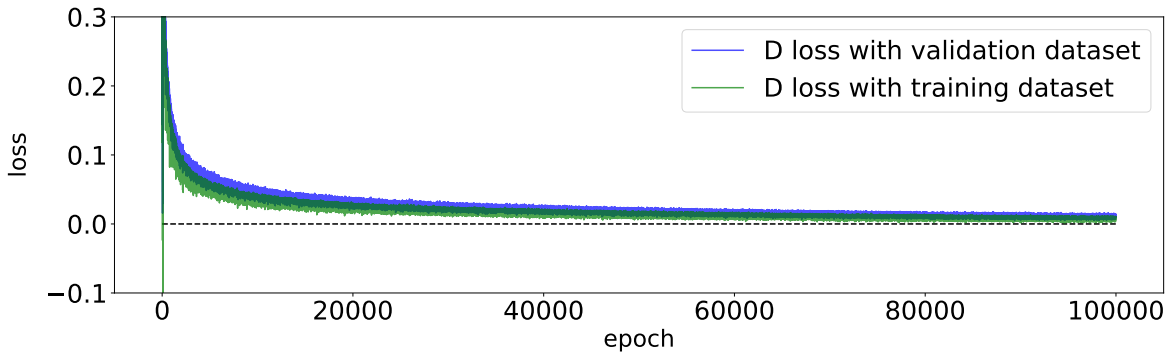

Figure 4. Critic (discriminator) Wasserstein loss as a function of the epoch, for training and the validation datasets during the Experiment1 training.

The figure 9 depicts several slices, along the three axis, of the reconstructed images (208 keV) with reference and GAN method for the first experiment. By using the spherical regions of activities, we computed the relative difference between the mean number of counts in the regions obtained from the reference and GAN reconstructed images and obtained -1.7%, 2.0% and -2.6% respectively for the background, ×20 and ×50 spheres.

For the second experiment, the figure 10 displays the results of the reconstructed images both from reference and GAN based simulation. Because of the well known

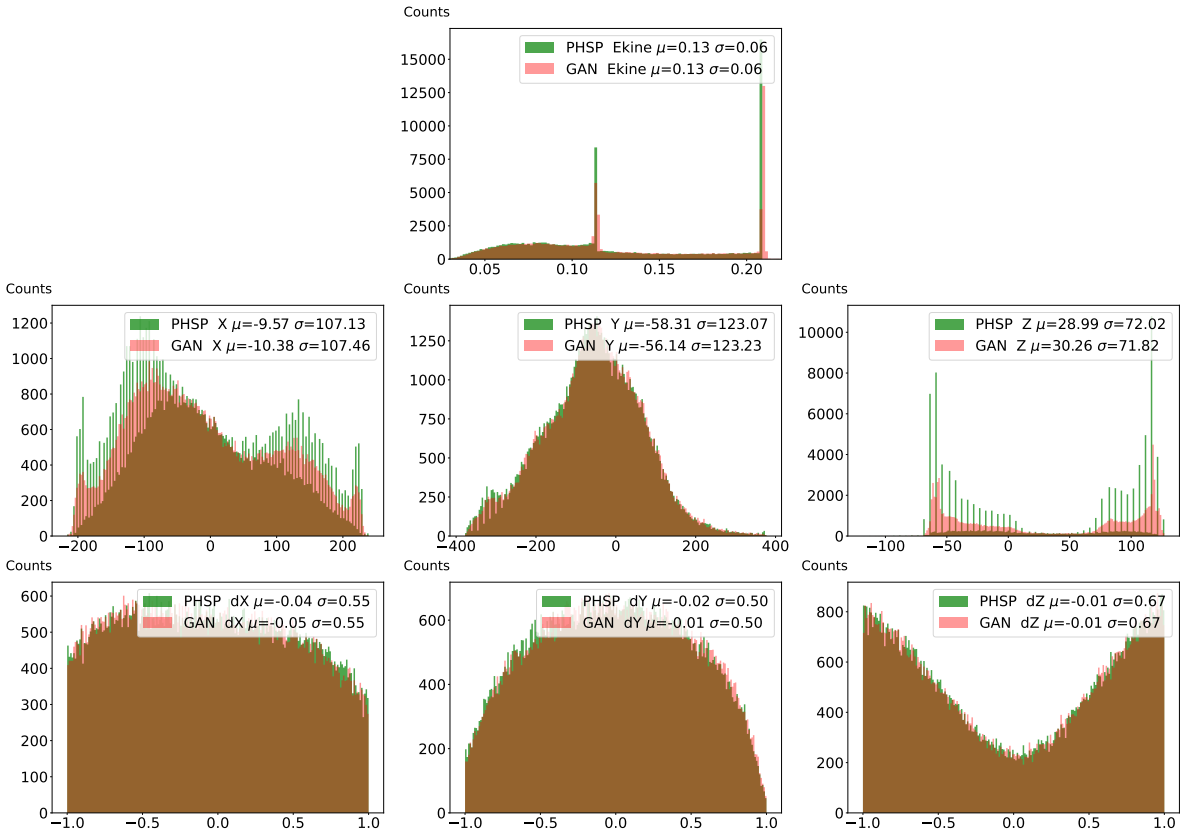

Figure 5. Marginal histograms obtained from the reference phase space and generated from the generator G of the GAN.

confounding physical effects such as partial volume effect due to limited spatial resolution, scatter, and photon attenuation of SPECT acquisition, the input activity (shown in figure 2) is actually expected to differ from any reconstructed images regardless of the simulation method. Therefore, the input activity image is not intended to be compared with the reconstructed ones. What counts here are the differences between the reconstructed images based on reference analog Monte Carlo simulation (top row) and on GAN (bottom row).

The table 3 at left displays the Hellinger distance between all the 8 gradient penalties presented in table 1 for various λ values. At right, the table shows Hellinger distance for several training dataset sizes, when using Square Hinge with $\lambda = 10$. Note the values of the distance were scaled by 100 for clarity. The values can only be interpreted relatively to each others. The color scale of the table is chosen to illustrate the range, from green (min) to red (max) and the figure 11 illustrates the difference between some of the generated images according to their Hellinger distance values with the reference image (the chosen images are circled in blue in the table 3).

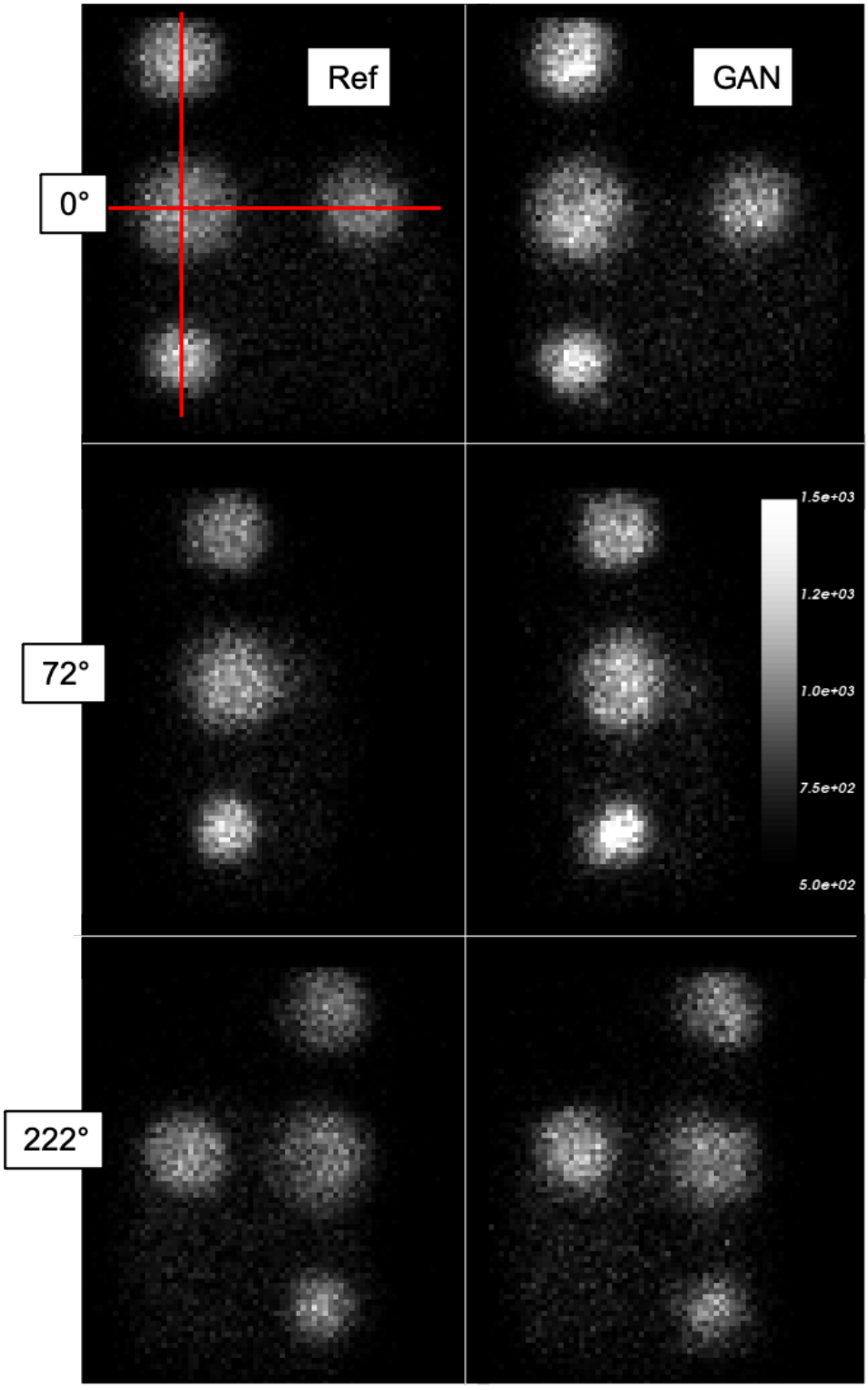

Figure 6. Projections from the reference and the GAN simulation, for three different angles, for the 208 keV energy window. The two red lines indicate where the profiles of the figure 8 are extracted.

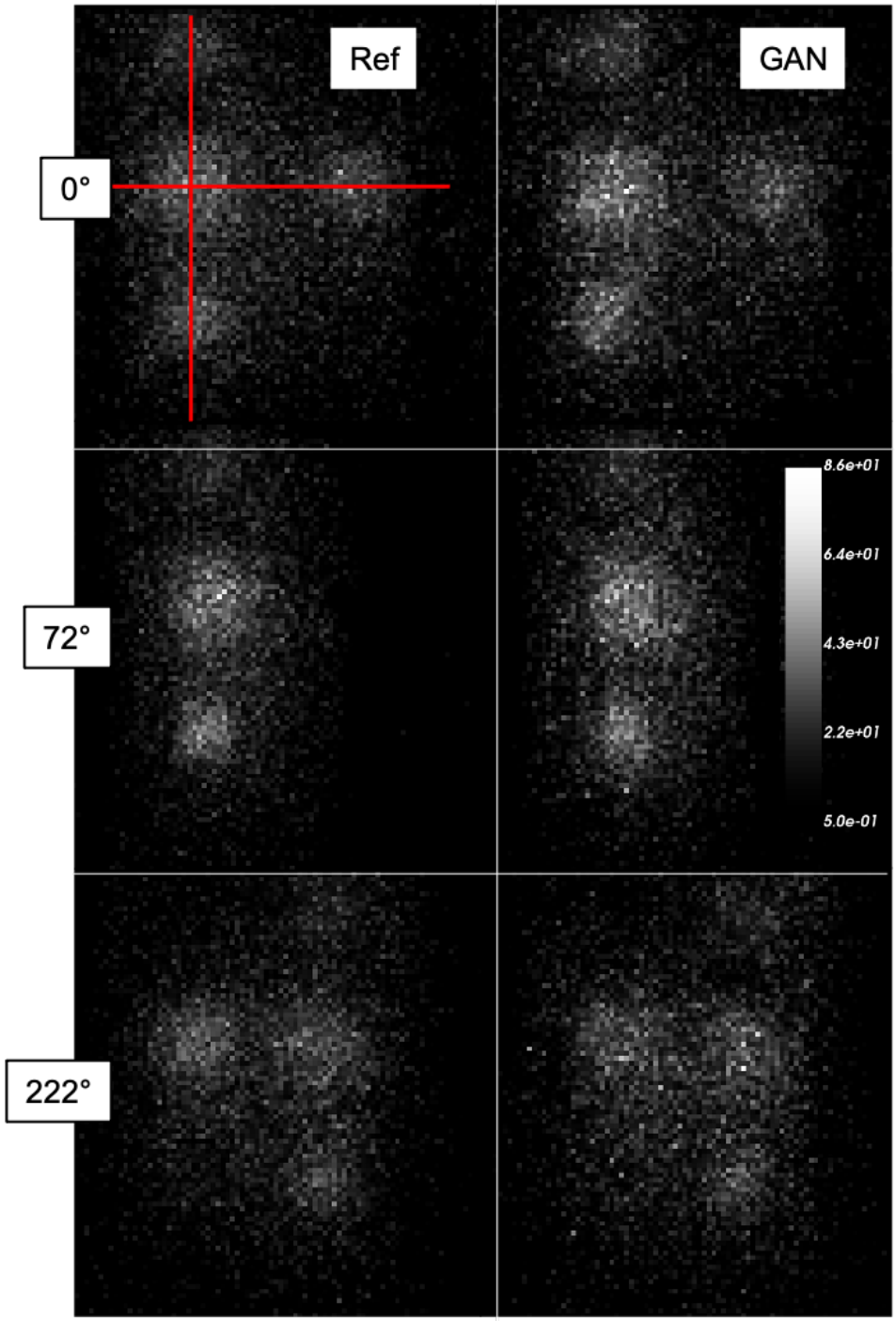

Figure 7. Projections from the reference and the GAN simulation, for three different angles, for the scatter1 (96-104 keV) energy window. The two red lines indicate where the profiles of the figure 8 are extracted.

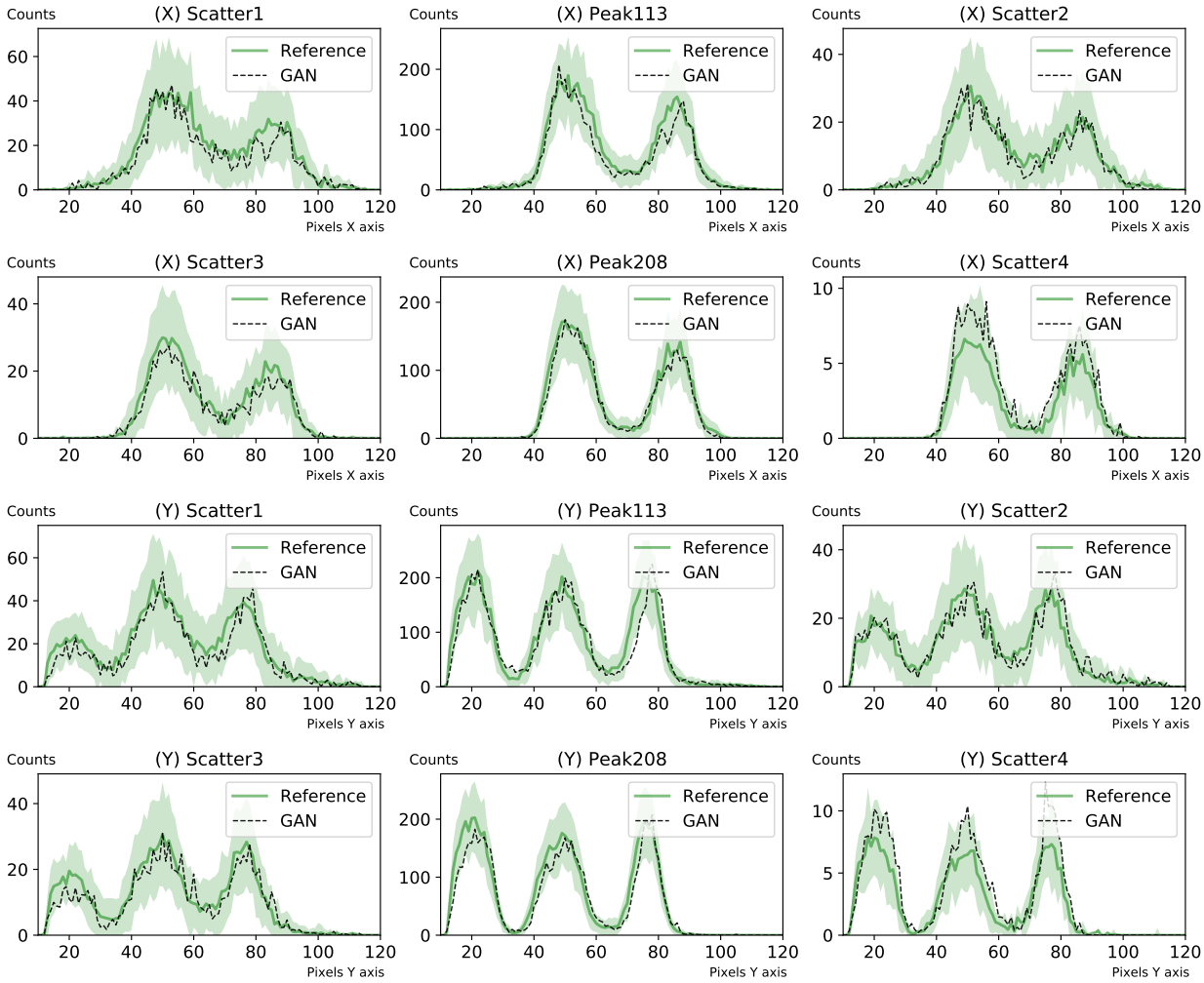

Figure 8. Profiles comparison for all six energy windows between reference simulation and GAN. Error bands are 2 times the standard deviation (95.5% confidence interval) obtained from the reference simulations. First two rows are horizontal profiles along X axis and last two rows are vertical profiles along Y axis. Note that the vertical axis (counts) are different for each energy window.

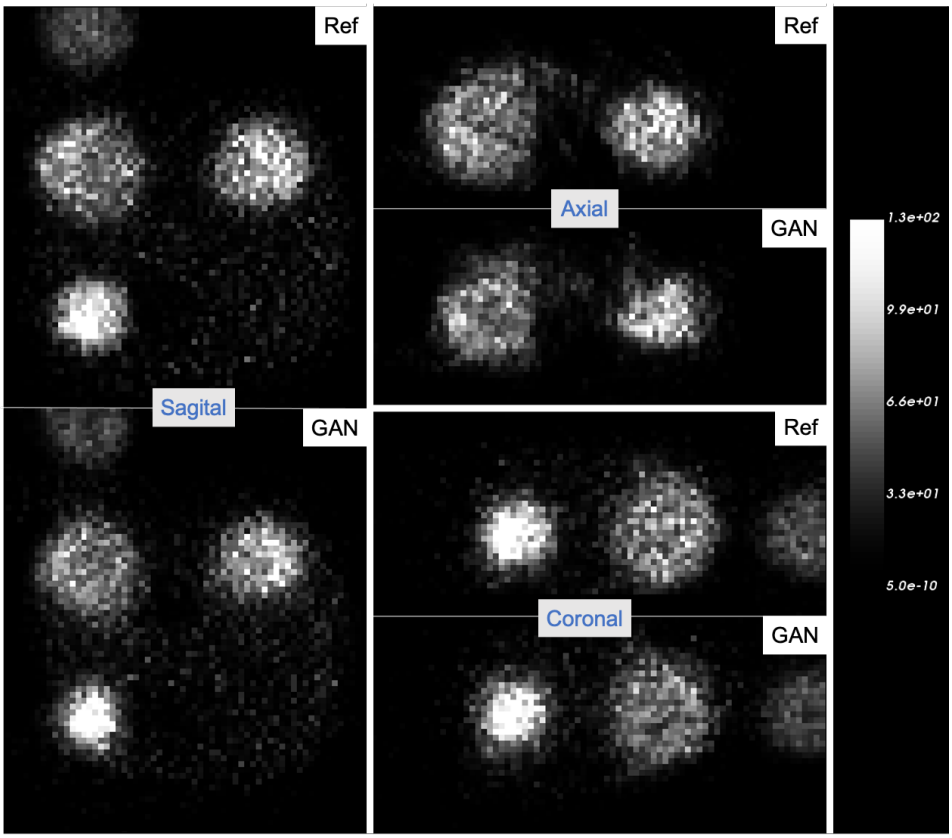

Figure 9. Slices of the 3D reconstructed images of experiment1 (208 keV) with reference and GAN methods.

λ	L1 LS	L2 LS	Linf LS	L1 Hinge	L2 Hinge	Linf Hinge	0-GP	Sq Hinge
0	2.98	3.20	2.79	4.82	4.29	3.84	3.03	2.97
0.05	2.74	2.72	2.84	7.54	4.85	6.07	3.09	2.86
0.10	2.75	2.82	2.71	3.51	3.83	4.21	2.73	2.77
0.5	2.94	2.89	2.68	2.73	2.84	2.92	2.98	2.79
5.0	6.46	2.78	2.73	2.79	2.67	2.78	2.96	2.63
10	7.57	2.81	2.90	2.86	2.72	2.75	2.72	2.62
20	12.81	3.08	4.03	2.86	2.70	2.69	2.63	2.65
50	42.04	5.37	4.44	2.87	2.91	2.69	2.80	2.66

decays	gamma	Hell dist
2.0E+09	2.9E+08	2.64
1.0E+09	1.4E+08	2.62
5.0E+08	7.2E+07	2.81
1.0E+08	1.4E+07	2.77
5.0E+07	7.2E+06	2.65
2.0E+07	2.9E+06	2.78
1.00E+07	1.4E+06	2.91

Table 3. Table at left: Hellinger distances between reference and GAN generated images for 8 different gradient penalty functions and 8 values of λ for the Experiment2. Value circled in blue are the selected examples of figure 11. Table at right: Hd for various training dataset sizes, expressed in decays and number of corresponding gammas, with Square Hinge and $\lambda = 10$.

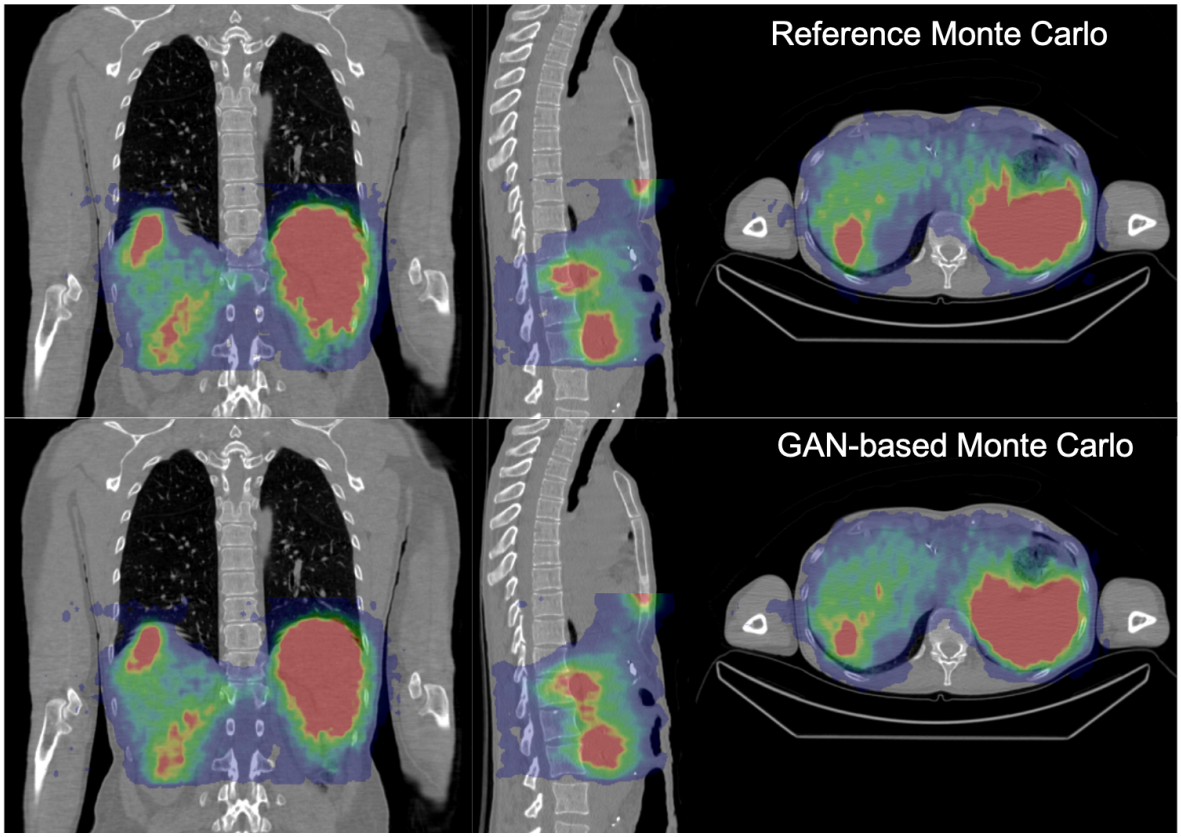

Figure 10. Reconstructed tomography SPECT overlaid on patient CT slides. Upper row: reconstruction performed with the projections obtained from the reference simulation plus ARF-nn; lower row: projections obtained via the combined GAN/ARF-nn method.

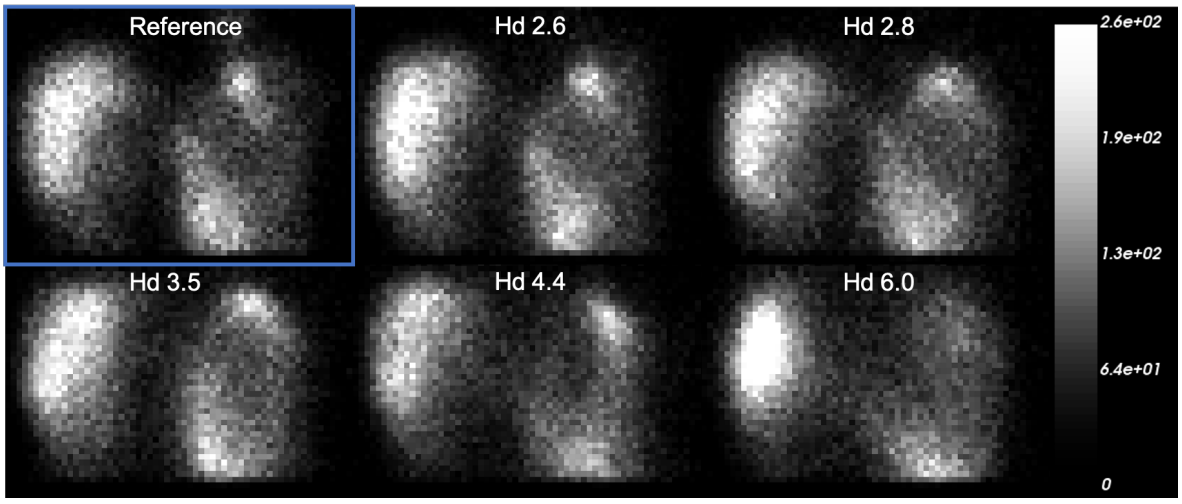

Figure 11. Examples of projection images (208 keV peak energy window), generated by the reference method (top-left) and with the GAN, for various Hellinger distances encountered in table 3 in order to visually appreciate the loss of quality associated with increasing distance. The color scale is the same for all images.

5. Discussion

The total computation time of the proposed method is composed of 1) the time needed to generate the learning dataset T_{MC} via Monte Carlo simulation 2) the GAN training time T_{train} and 3) the final generation time with the combined method $T_{GAN} + T_{ARF}$. In case of the reference method, only the Monte Carlo simulation and the ARF are relevant. In the following, we discuss these different contributions except T_{ARF}, which is the same for both methods and not detailed here.

Computing time analysis of T_{MC}. The computation time of the reference simulation depends on several parameters (energy cuts, type of physics list etc.), but the main one is the resolution of the voxelized CT volume. The computation speed is respectively 7300, 5900, 5000, 3500 and 1900 PPS (particles per seconds) for 5, 4, 3, 2 and 1 mm voxel size. It thus requires between 0.8 and 3 days of computation time to simulate 4.6×10^8 particles per projection and between 43 and 165 CPU-days (24 hours of computation time) for the SPECT acquisition comprised of 60 projections. Only a single simulation is necessary to generate the training dataset and the GAN is then used 60 times to generate all projections. Generating the training dataset is slightly slower (10%) than the reference simulation of one projection due to the time needed to write the particles in the phase-space to disk. We used a learning dataset of 10^9 decays corresponding to 1.7×10^8 emitted gammas, which took between 7 and 31 hours according to the voxel size. The resulting phase space file had 4 GB.

Learning gamma distributions with GAN (T_{train}). Training of the GAN is an iterative optimization process that depends on a large number of parameters. The exact influence of each parameter on the final accuracy remains difficult to assess. The size of the network is larger than of those used to learn a Linac phase space in [12]. The number of layers did not have a large influence on the results. Gradient penalty was required and led to significantly better results than with the weight clipping method. The penalty weight λ is not easy to determine and depends on the type of penalty. No systematic differences were observed between the different penalty flavors (left table 3). The penalty L1-LS seems not as good as the others, whatever the value of λ, and Squared Hinge seems a bit better than the others for a larger range of λ values. This table gives an indication of the sensitivity of λ. It is reasonable to expect that the accuracy of the trained GAN depends on the size of the training dataset. The initial training was performed with 1.5×10^8 particles (10^9 decays). The influence of using more or fewer particles for training is displayed in the right part of table 3: the discrepancy between reference and GAN based images begins to increase when fewer than 7×10^7 particles are considered (385 MB file size). On the other hand, from a certain number of gammas upwards the accuracy of the GAN appears to remain relatively unaffected. We underline here that the statistical noise in GAN generated images mainly depends on the number of generated particles, i.e. the size of the input to G, and not on the size of the training

dataset. A detailed study of noise properties of GAN generated images was beyond the scope of this work and will require further investigation.

A large batch size $>10^5$ is also required to approximate the complex 7-dimensional distribution. Theoretical arguments have been given to have a larger number of critic (D) updates than generator (G) updates during one epoch. However, we did not obtain any better results with more D updates. The learning rates were also set experimentally. The learning procedure is stochastic and slight differences were observed between two trainings. Hence, the provided set of parameters that have been chosen according to theoretical considerations in the literature lead to adequate result but it is probably not optimal. The training time (using GPU) was about 23 minutes for 10^4 epoch, leading to less than about 4 hours for 10^5.

It is not known yet what the optimal gradient penalty function and optimal λ value are and how they depend on the setting of the simulation. Better or faster training procedures may be obtained in the future when more knowledge about GAN will be available. GAN is still a very active field of research and new developments and theoretical studies of training behavior are still ongoing. As an example, the recent work described in [21] studied and compared several forms of gradient norm penalty strategies. With the method presented in the current work, a new GAN must be trained for each new patient or activity distribution. Transfer learning with GAN architecture [33] could be a starting point to address this. As in other deep learning applications, it is expected that pre-trained GAN models could be used as starting point in order to speed up training and improve the performance. A more complex way to replace Monte Carlo by deep learning would be to use the patient CT and source distribution and train a network to predict the exit phase space. The training would not need to be repeated for each patient. Whether this could be achieved e.g. via conditional GAN requires further investigation.

GAN and image generation time T_{GAN}. The generation of the final images with the trained GAN consists in 1) generating the gamma via the GAN, 2) computing the intersection with the ARF plane and 3) apply the ARF-nn. The combined computation time of those 3 steps, was performed at approximately 600,000 particles per seconds (PPS) leading to about 12 min for one projection or 12 hours for the whole 60 projections. With the proposed method, the computation time is independent of the voxel size (except for generating the training dataset). The training is only done once, so if the image detection model is modified (e.g. to study the imager design), only the GAN image generation part should be performed with a different ARF-nn model.

Using a different radionuclide than ^{177}Lu should modify only slightly those numbers. For example, with ^{99m}Tc that decays to lower energy gammas of 140.5 KeV, we can expect more scatter in the anatomical images so a slightly larger ratio between emitted and exited gamma, and slightly improved speed up.

Gamma tracking. It is worth mentioning that the currently implemented SPECT Monte Carlo simulation is not perfectly efficient because it simulates gammas from an isotropic source distribution while only those are eventually considered for image formation which are emitted into the solid angle defined by the SPECT imager. Therefore, part of the computation time is spent for tracking particles that will never really reach the detector. Indeed, similar simulation is simply repeated for each projection angle. A way to quantify the efficiency of a SPECT simulation would be to determine the ratio of tracked gammas to those actually used for image formation. A large ratio would mean inefficient simulation and a ratio of one would be ideal. In case of the reference simulation, the described inefficiency impacts the term T_{MC}. Forced Detection techniques already available in SIMIND [34] or Gate [23] are a good way to improve the simulation efficiency because each photon is directed towards the detector and therefore contribute to the projection formation. When simulating a complete SPECT acquisition (i.e. rotation of the imaging device), it could also be feasible to check each gamma's coordinates when exiting the patient volume and then select an imager position (i.e. position of the ARF plane) which the gamma would actually reach. No tools are currently available in GATE/Geant4 to realise this and, while feasible, it would require a large development to deal with overlapping planes and to handle situations where a photon trajectory would cross several detector planes.

In the proposed GAN-based method, it is the term T_{GAN} which is larger than it would ideally be because the GAN currently generates particles in all directions regardless of the placement of the ARF plane. Indeed, the same GAN+ARF step is executed repeatedly, i.e. once for each projection. This could be improved in future work either by employing a similar concept as above to dynamically select a suitable ARF position or by training a conditional GAN [35] to impose constraints on the generated gammas. We considered that this improvement is of the same order of magnitude here and for the reference Monte Carlo method.

Other considerations. In terms of simulation accuracy, we quantified the difference between images generated via conventional simulation and with GAN generated gammas. According to our results, images with low error compared to the reference seem feasible. However, which level of difference is acceptable ultimately depends of the application. Further works are needed to better understand the limitation and potential bias of the method.

It is difficult to provide a fair timing comparison as part of the process is performed by neural networks and thus use GPU, while all other computations are CPU only. Of course, parallelisation of Monte Carlo on multiple CPUs and GPUs is possible (and advocated) as every event processing is independent. As an example, hundreds of parallel CPUs were used here for the reference simulation, leading to a few days computation time, and several parallel GPUs were employed for the final computation which thus took a few hours. Finally, further time gain is expected when the code to train and apply the GAN is fully optimized. This efficiency is still far from a direct

fully dedicated GPU Monte Carlo code [5] reaching 3200 emitted million photons/s that remains perfectly adapted for reconstruction methods. The proposed principle, however, is more general and does not rely on forced interaction or adjusted cross-sections. It may therefore be adapted to other types of imaging systems. It could be particularly useful when designing of a new imaging system, to study acquisition parameters, to evaluate scatter correction techniques, or for matrix system computation.

As a side effect, the proposed mask method allows to solve the issue of volume overlap in GATE simulations where the rotating SPECT head intersects the patient volume for some angles of rotation and system design. In Geant4, the Monte Carlo library underlying GATE, the behavior of the particle tracking algorithm is undefined and leads to incorrect results when volumes are overlap.

Limitations. In the current experiments, the particles generated by the generator G are prolonged in straight lines to the detectors. If any object is present between the patient and the detectors, it is thus ignored. However, the generator may be used as a conventional source in order to let the Monte Carlo simulation engine track the generated particle through potential intermediate objects. In absence of such object, the Monte Carlo engine anyhow transports along straight lines. Currently, the GAN provides no information about the particle time and detector dead time is thus ignored. Time may potentially be added to the training dataset as an additional dimension and learnt with the generator. It will be for example required for exploiting the proposed method for other types of imaging systems that require event time coincidences, such as PET or Compton Camera.

6. Conclusion

In this work, we investigated the feasibility to learn complex particle distributions with GAN for SPECT simulation in order to replace a phase space with neural network generator during Monte Carlo simulation. Our results show that this is feasible and that simulations can be speed up by two or three orders of magnitude according to the configuration. Further work remains to be performed to better characterize the statistical properties of GAN generated phase space.

Acknowledgments

This work was performed within the framework of the SIRIC LYriCAN Grant INCa-INSERM-DGOS-12563, the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program “Investissements d’Avenir” (ANR- 11-IDEX-0007) operated by the ANR, and the POPEYE ERA PerMed 2019 project (ANR-19-PERM-0007-04). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This work was granted access to the

HPC resources of IDRIS under the allocation 2019-101203 made by GENCI (Jean Zay computing center).

References

[1] Jeremy M. C. Brown. In-silico optimisation of tileable Philips digital SiPM based thin monolithic scintillator detectors for SPECT applications. *Applied Radiation and Isotopes*, page 109368, October 2020.

[2] Roberto Massari, Annunziata D'Elia, Andea Soluri, and Alessandro Soluri. Super Spatial Resolution (SSR) method for small animal SPECT imaging: A Monte Carlo study. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 982:164584, December 2020.

[3] B. Auer, N. Zeraatkar, S. Banerjee, J. C. Goding, L. R. Furenlid, and M. A. King. Preliminary investigation of a Monte Carlo-based system matrix approach for quantitative clinical brain 123I SPECT imaging. In *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, pages 1–2, November 2018.

[4] X Song, W P Segars, Y Du, B M W Tsui, and E C Frey. Fast modelling of the collimator–detector response in Monte Carlo simulation of SPECT imaging using the angular response function. *Physics in Medicine and Biology*, 50(8):1791–1804, April 2005.

[5] T. Rydén, J. Heydorn Lagerlöf, J. Hemmingsson, I. Marin, J. Svensson, M. Båth, P. Gjertsson, and P. Bernhardt. Fast GPU-based Monte Carlo code for SPECT/CT reconstructions generates improved 177Lu images. *EJNMMI Physics*, 5(1):1, December 2018.

[6] D Sarrut, N Krah, J N Badel, and J M Létang. Learning SPECT detector angular response function with neural network for accelerating Monte-Carlo simulations. *Physics in Medicine & Biology*, 63(20):205013, October 2018.

[7] Pedro Andreo. Monte Carlo simulations in radiotherapy dosimetry. *Radiation Oncology*, 13(1):121, 2018.

[8] Lorenzo Brualla, Miguel Rodriguez, Josep Sempau, and Pedro Andreo. PENELOPE/PRIMO-calculated photon and electron spectra from clinical accelerators. *Radiation Oncology*, 14(1):6, 2019.

[9] I Chabert, E Barat, Thomas Dautremer, Thierry Montagu, M Agelou, A Croc de Suray, JC Garcia-Hernandez, S Gempp, M Benkreira, L De Carlan, et al. Development and implementation in the Monte Carlo code PENELOPE of a new virtual source model for radiotherapy photon beams and portal image calculation. *Physics in Medicine & Biology*, 61(14):5215, 2016.

[10] L Grevillot, T Frisson, D Maneval, N Zahra, J-N Badel, and D Sarrut. Simulation of a 6 MV Elekta Precise Linac photon beam using GATE/GEANT4. *Physics in Medicine and Biology*, 56(4):903–918, February 2011.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[12] David Sarrut, Nils Krah, and Jean-Michel Letang. Generative adversarial networks (GAN) for compact beam source modelling in Monte Carlo simulations. *Physics in Medicine and Biology*, August 2019.

[13] W. Schneider, T. Bortfeld, and W. Schlegel. Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions. *Phys Med Biol*, 45(2):459—478, 2000.

[14] David Sarrut and Laurent Guigues. Region-oriented CT image representation for reducing computing time of Monte Carlo simulations: Voxelized geometry with GEANT4. *Medical Physics*, 35(4):1452–1463, March 2008.

[15] Simon R. Cherry, James A. Sorenson, and Michael E. Phelps, editors. *Physics in Nuclear Medicine*. W.B. Saunders, Philadelphia, January 2012.

[16] Romulo Pinho, Vivien Delmon, Jef Vandemeulebroucke, Simon Rit, and David Sarrut. Keuhkot: A Method for Lung Segmentation. In *MICCAI - LOLA11 Challenge*, 2011.

[17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, December 2017.

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs. In *NIPS 2017*, December 2017.

[19] Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of Wasserstein GANs. *arXiv:1709.08894 [cs, stat]*, March 2018.

[20] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving Generalization and Stability of Generative Adversarial Networks. In *ICLR2019*, February 2019.

[21] Alexia Jolicoeur-Martineau and Ioannis Mitliagkas. Connections between Support Vector Machines, Wasserstein distance and gradient-penalty GANs. *arXiv:1910.06922 [cs, stat]*, October 2019.

[22] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[23] Thomas Cajgfinger, Simon Rit, Jean Michel Létang, Adrien Halty, and David Sarrut. Fixed forced detection for fast SPECT Monte-Carlo simulation. *Physics in Medicine & Biology*, 63(5):055011, 2018.

[24] David Sarrut, Manuel Bardiès, Nicolas Boussion, Nicolas Freud, Sébastien Jan, Jean-Michel Létang, George Loudos, Lydia Maigne, Sara Marcatili, Thibault Mauxion, et al. A review of the use and potential of the GATE Monte Carlo simulation code for radiation therapy and dosimetry applications. *Medical physics*, 41(6Part1):064301, June 2014.

[25] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso, E. Bagli, A. Bagulya, S. Banerjee, G. Barrand, B.R. Beck, A.G. Bogdanov, D. Brandt, J.M.C. Brown, H. Burkhardt, Ph. Canal, D. Ott, S. Chauvie, K. Cho, and H. Yoshida. Recent developments in GEANT4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835:186–225, 2016.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *NEURIPS 2019*, page 12, 2019.

[27] K. Assié, I. Gardin, P. Véra, and I. Buvat. Validation of the Monte Carlo simulator GATE for indium-111 imaging. *Physics in Medicine and Biology*, 50(13):3113–3125, July 2005.

[28] Antoine Robert, Simon Rit, Thomas Baudier, Julien Jomier, and David Sarrut. 4D respiration-correlated whole-body SPECT reconstruction. In *2019 IEEE NSS-MIC*, 2019.

[29] S. Rit, M. Vila Oliva, S. Brousmiche, R. Labarbe, D. Sarrut, and G. C. Sharp. The Reconstruction Toolkit (RTK), an open-source cone-beam CT reconstruction toolkit based on the Insight Toolkit (ITK). *Journal of Physics: Conference Series*, 489:012079, March 2014.

[30] A.R. De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Transactions on Medical Imaging*, 14(1):132–137, March 1995.

[31] K. F. Koral, F. M. Swailem, S. Buchbinder, N. H. Clinthorne, W. L. Rogers, and B. M. Tsui. SPECT dual-energy-window Compton correction: Scatter multiplier required for quantification. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, 31(1):90–98, January 1990.

[32] G. L. Zeng, C. Bai, and G. T. Gullberg. A projector/backprojector with slice-to-slice blurring for efficient three-dimensional scatter modeling. *IEEE transactions on medical imaging*, 18(8):722–732, August 1999.

[33] Yaël Frégier and Jean-Baptiste Gouray. Mind2Mind : Transfer learning for GANs.

arXiv:1906.11613 [cs, stat], June 2019.

[34] Johan Gustafsson, Gustav Brolin, and Michael Ljungberg. Monte Carlo-based SPECT reconstruction within the SIMIND framework. *Physics in Medicine & Biology*, 63(24):245012, December 2018.

[35] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, November 2014.